Introduction to JSFL

Who am I ?

George Profenza

- Flash Developer at **disturb media**
- **Creative Computing** student at Goldsmiths,
University of London

What will we talk about today ?

We will learn how to get more out of the Flash IDE

Who is this talk for ?

Since we're at
The London Flash Developers and Designers Meetup Group
it is for both **Developers** and **Designers**.

What will we talk about today ?

We will learn how to get more out of the Flash IDE

Who is this talk for ?

Since we're at
The London Flash Developers and Designers Meetup Group
it is for both **Developers** and **Designers**.

How can we get out more out of the Flash IDE ?

What will we talk about today ?

We will learn how to get more out of the Flash IDE

Who is this talk for ?

Since we're at
The London Flash Developers and Designers Meetup Group
it is for both **Developers** and **Designers**.

How can we get out more out of the Flash IDE ?

Using JSFL.

What is JSFL ?

What is JSFL ?

JSFL is short for **JavaScript Flash**

What is JSFL ?

JSFL is short for **JavaScript Flash**

JSFL is the **scripting language** for the **Flash IDE**

This means we will be writing a bit of Javascript to control the Flash IDE.

That's about it!

Why would we want to do that ?

Why would we want to do that ?

"He that would perfect his work must first sharpen his tools." Confucius (551-479)

Why would we want to do that ?

The no.1 reason I use jsfl is to
GET READ OF TEDIOUS REPETITIVE TASKS !

Why would we want to do that ?

The no.1 reason I use jsfl is to
GET READ OF TEDIOUS REPETITIVE TASKS !

The no.2 reason is it allows a bit of freedom to
create your own tools and make cool stuff.

How many times have you done things like this:
- add stop()
- duplicate layer
- guide+hide a layer
- reset the registration of a MC by aligning its content centre/TL
- flip horizontal/vertical

...
?

How many times have you done things like this:
- add stop()
- duplicate layer
- guide+hide a layer
- reset the registration of a MC by aligning its content centre/TL
- flip horizontal/vertical

...
?

How much time did you spend doing this ?

How many times have you done things like this:
- add stop()
- duplicate layer
- guide+hide a layer
- reset the registration of a MC by aligning its content centre/TL
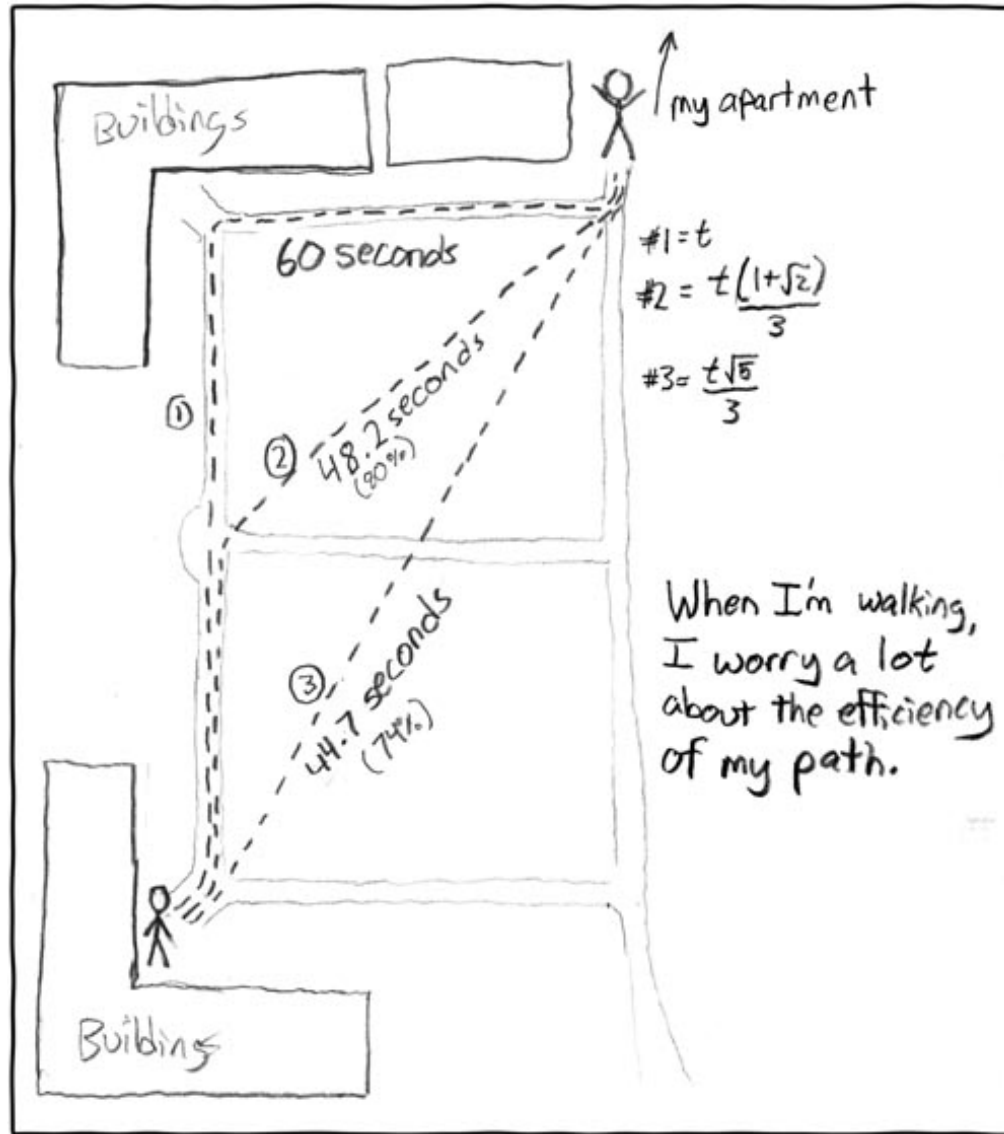- flip horizontal/vertical

...
?

How much time did you spend doing this ?
Could it be more efficient ?

Randall Monroe, http://xkcd.org/85/

Some basic examples:

[LazyBoy Panel by me](#)

[Copy Fill as AS3 by Tink](#)

[Ouput Flash Filters,Record Stage Animation, etc.](#)

by Slavomir Durej

Some animation tools (some of them oldschool, but handy) :

.
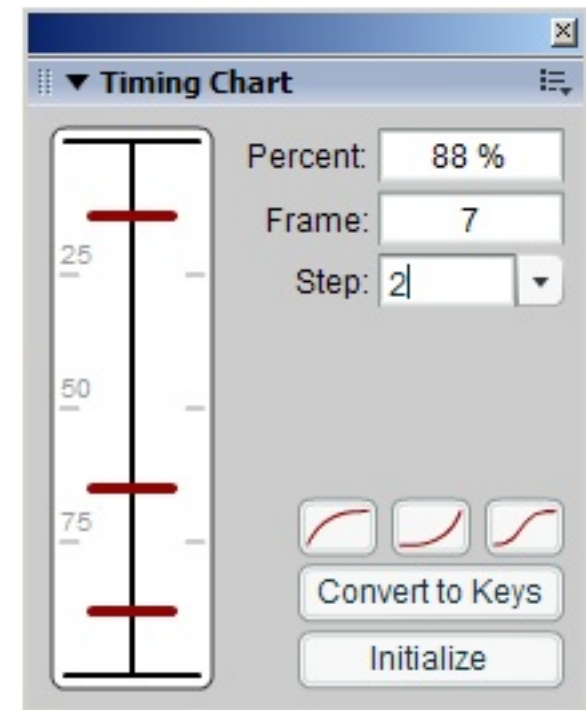http://www.dave-logan.com/weblog/?p=46
http://www.toonmonkey.com/extensions.html
http://www.animonger.com/freetools.html

Autocolor by Dave Logan

Timing Chart by ToonMonkey

Great CS4 extensions by Justin Putney (ajarproductions)

Before

After

There quite a few handy things to be made:

- **commands** - to bind to keyboard shortcuts
- **panels** - windowSWF/swfPanel
- **tools**
- other-oldschool (behaviors - as2.0 only, timeline effects*)
- **new** - procScripts - Deco Tool, etc

*timeline effects were present in Flash up until version 9(CS 3)

There quite a few handy things to be made:

- commands - to bind to keyboard shortcuts
- panels - windowSWF/swfPanel
- tools - oldschool (behaviors - as2.0 only, timeline effects*)
- new - procScripts - Deco Tool

Ok, now we know JSFL is.
Time to get something done:

Tutorial 1

Tutorial 1
- keep an eye out for a repetitive task

Tutorial 1
- keep an eye out for a repetitive task
- our goal is to progress/learn from our past > History Tool
    - finds steps that aren't marked' with a red X

Tutorial 1
- keep an eye out for a repetitive task
- our goal is to progress/learn from our past > History Tool
    - finds steps that aren't marked' with a red X
    - selects the steps needed > click the floppy disk icon
(Save selected steps as command)

hey! no code! :)

Tutorial 1
- keep an eye out for a repetitive task
- our goal is to progress/learn from our past > History Tool
    - finds steps that aren't marked' with a red X
    - selects the steps needed > click the floppy disk icon
(Save selected steps as command)

hey! no code! :)

Now you can assign a keyboard shortcut
(Flash > Keyboard Shortcuts)

Tutorial 2
  Inspecting what is happening
    - open a saved jsfl file

Tutorial 2

Inspecting what is happening
- open a saved jsfl file
- look at the code - some of it is 'readable'

Tutorial 2
   Inspecting what is happening
    - opening a jsfl file
    - look at the code - some of it is 'readable'
    - repeat (what you do when you don't understand something)

Tutorial 2
   Inspecting what is happening
      - opening a jsfl file
      - look at the code - some of it is 'readable'
      - repeat (what you do when you don't understand something)

History Panel
      - select the Panel's context menu(
      - view (from the Panel's context menu) >
                        arguments in panel /
                        javascript in panel   /
                        arguments in tooltip, etc.

More JSFL

Commands

Mostly you will be working with the **document**, accessing properties of **elements**(like MovieClips, Drawing Shapes, etc.)

You can either use the current **selection**, either use access the **timeline** and **layers** to get to the element you want to modify.

Before we move on, I need to point out that in CS4 the documentation in not included in the IDE, but it present online.

It might be handy to download it as a PDF for offline reference.

The **document**:

```
var doc = fl.getDocumentDOM();
```

The **document**:

```
var doc = fl.getDocumentDOM();
```

The **selection**

```
fl.trace(doc.selection);//result is array
```

You can select **everything** if you want:

```
doc.selectAll();
```

Or **nothing**(clear selection):

```
doc.selectNone();
```

Accessing the selection

```
var doc = fl.getDocumentDOM();//get the current document ref.
var selection = doc.selection;//get the selection

var elementsNum = selection.length;//store this for counting*
for(var i = 0 ; i  < elementsNum ; i++){
    fl.trace(selection[i]);
}
```

Accessing the selection

```
var doc = fl.getDocumentDOM();//get the current document ref.
var selection = doc.selection;//get the selection

var elementsNum = selection.length;//store this for counting*
for(var i = 0 ; i  < elementsNum ; i++){
    fl.trace(selection[i].name+'.x = ' + selection[i].x+';');
}
```

Accessing the selection, generating XML

The JSFL Engine was updated to Javascript 1.6 in Flash CS3, which also means you get to use XML and E4X!
For more details see [Robert Penner's post](#) on this.

```
var doc = fl.getDocumentDOM();//get the current document ref.
var selection = doc.selection;//get the selection
var layout = <layout />;//create the root node for our xml

var elementsNum = selection.length;//store this for counting*
for(var i = 0 ; i  < elementsNum ; i++){
    layout.appendChild(<element />);//add an element node
    layout.element[i].@name = selection[i].name;//setup attributes
    layout.element[i].@x = selection[i].x;
    layout.element[i].@y = selection[i].y;
}
fl.trace(layout); //let's see some output
```

Accessing the selection, generating XML

The JSFL Engine was updated to Javascript 1.6 in Flash CS3, which also means you get to use XML and E4X!
For more details see [Robert Penner's post](#) on this.

```
var doc = fl.getDocumentDOM();//get the current document ref.
var selection = doc.selection;//get the selection
var layout = <layout />;//create the root node for our xml

var elementsNum = selection.length;//store this for counting*
for(var i = 0 ; i  < elementsNum ; i++){
    layout.appendChild(<element />);//add an element node
    layout.element[i].@name = selection[i].name;//setup attributes
    layout.element[i].@x = selection[i].x;
    layout.element[i].@y = selection[i].y;
}
fl.trace(layout); //let's see some output
```

Accessing the selection, generating XML

The JSFL Engine was updated to Javascript 1.6 in Flash CS3, which also means you get to use XML and E4X!
For more details see [Robert Penner's post](#) on this.

```
var doc = fl.getDocumentDOM();//get the current document ref.
var selection = doc.selection;//get the selection
var layout = <layout />;//create the root node for our xml

var elementsNum = selection.length;//store this for counting*
for(var i = 0 ; i  < elementsNum ; i++){
    layout.appendChild(<element />);//add an element node
    layout.element[i].@name = selection[i].name;//setup attributes
    layout.element[i].@x = selection[i].x;
    layout.element[i].@y = selection[i].y;
}
fl.clipCopyString(layout);//now it's in the clipboard!
```

Accessing the selection, generating XML

Tiny gotcha!

The autocomplete will display

```
fl.copyClipString(string);
```
not
fl.clipCopyString(string);Trust the documentation, not the IDE on this one

Accessing the selection, generating XML, saving to a file

```
var doc = fl.getDocumentDOM();//get the current document ref.
var selection = doc.selection;//get the selection
var layout = <layout />;//create the root node for our xml

var elementsNum = selection.length;//store this for counting*
for(var i = 0 ; i  < elementsNum ; i++){
    layout.appendChild(<element />);//add an element node
    layout.element[i].@name = selection[i].name;//setup attributes
    layout.element[i].@x = selection[i].x;
    layout.element[i].@y = selection[i].y;
}
fl.trace(FLfile.write("~/layout.xml",layout));//beware!
```

FLfile warnings:
1. setup the URI properly('file:///c:'...on PC, '/Work/prj'...on OSX
2. we trace the result, because you can write files only if you uses has the rights to do so!
3. Use it wisely, there's no UNDO if you mess up files!

You can use JSFL with actionscript 3.0 by writing a Flash Panel (aka swfPanel, Window SWF)

In actionscript all you need to do is call:

```
MMExecute("your javascript here");
```

e.g.

```
MMExecute("alert('Hello from AS3!')");
```

The function returns a String which is the result of the JSFL operation(s), handy for debugging.

A very good video tutorial on Lee Brimelow's gotoandlearn

Where to go from here:

Since you know scripting basics and how to access documentation, you can script away in other software.

If you look in CS4's extension manager, you see Photoshop, Illustrator, etc,
You can actually develop Flash Panels for those apps, work with AIR, etc.

Homework:
Look at other software on your machine, see if has a javascript API and print an alert :)

Thanks LSBU for hosting us!

Thank you for dropping by ^_^

George Profenza - disturb media